2

9349 P.03

Application No. 09/727,744 Amendment dated March 24, 2005 Reply to Office Action dated February 9, 2005

Amendments to the Claims

This listing of claims will replace all prior versions, and listings of claims in the application.

Listing of Claims

(Currently Amended) A method for processing a conditional jump 1 1. instruction in a pipelined instruction processor, the method comprising: 2 generating at least one status bit based on a digital value to be stored, the at least 3 4 one status bit relating to a particular condition of a conditional jump instruction and 5 specifying if the particular condition of the conditional jump instruction is satisfied or б not; 7 storing the digital value and the at least one status bit to a memory; and in response to a conditional jump instruction, reading from the memory the digital 8 value and the at least one status bit to determine if resolve whether the condition of the 9 conditional jump instruction is satisfied before the conditional jump instruction reaches 10 without having to submit the condition of the conditional jump instruction to an 11 12 arithmetic logic stage of the pipelines instruction processor. 1 2. (Cancel) 3. (Previously Presented) The method recited in claim 1, wherein the 1

2 of 14

at least one status bit is read from memory at the same time as the digital value.

l	4.	(Previously Presented)	The method recited in claim 1, wherein the
2	memory has o	one or more addressable location	ons, and the at least one status bit is stored at
3	the same addr	essable location as the corresp	onding digital value.
1	5.	(Previously Presented)	The method recited in claim 1, wherein the
2	at least one st	atus bit is set high if the digita	value is zero.
1	6.	(Previously Presented)	The method recited in claim 1, wherein the
2	at least one st	atus bit is set high if the digita	l value is a positive value.
1	7.	(Previously Presented)	The method recited in claim 1, wherein the
2	at least one st	atus bit is set high if the digita	l value is negative.
			.
1	8.	(Previously Presented)	The method recited in claim 1, wherein the
2	at least one st	atus bit is set high if the digita	l value is a non zero value.
1			
1	9.	(Previously Presented)	The method recited in claim 1, wherein the
2	at least one	status bit is set high based or	the value of the least significant bit of the
3	digital value.		
l	10.	(Currently Amended) In a pr	pelined instruction processor that executes
2	instructions in	ncluding conditional jump inst	ructions, one or more of the conditional jump

16

Application No. 09/727,744 Amendment dated March 24, 2005 Reply to Office Action dated February 9, 2005

3	instructions reading a digital value from memory to determine if the condition of the
4	conditional jump instruction is satisfied, the improvement comprising:
5	a status bit generator for generating at least one status bit based on a digital value,
6	the at least one status bit relating to a particular condition of a conditional jump
7	instruction and specifying if the particular condition of the conditional jump instruction
8	is satisfied or not;
9	storing means for storing the digital value and the at least one status bit to the
10	memory; and
11	conditional jump processing means, activated in response to the execution of a
12	conditional jump instruction, the conditional jump processing means reading from the
13	memory the digital value and the at least one status bit to resolve whether determine if the
14	condition of the conditional jump instruction is satisfied before the conditional jump
15	instruction reaches without having to submit the condition of the conditional jump
	·

11. (Cancel)

1 12. (Previously Presented) The pipelined instruction processor recited

instruction to an arithmetic logic stage of the pipelines instruction processor.

- 2 in claim 10, wherein the at least one status bit is read from the memory at the same time
- 3 as the digital value is read.

1	13.	(Previously Presented)	The pipelined instruction processor recited
2	in claim 10, v	wherein the memory has one	or more addressable locations, and the at least
3	one status bi	t is stored at the same add	dressable location as the corresponding digital
4	value.		
1	14.	(Previously Presented)	The pipelined instruction processor recited
2	in claim 10, v	wherein the at least one statu	s bit is set high if the digital value is zero.
1	15.	(Previously Presented)	The system recited in claim 10, wherein the
2	at least one st	atus bit is set high if the dig	ital value is a positive value.
		*	
1	16.	(Previously Presented)	The system recited in claim 10, wherein the
2	at least one st	tatus bit is set high if the dig	ital value is negative.
1	17.	(Previously Presented)	The system recited in claim 10, wherein the
2	at least one st	tatus bit is set high if the dig	ital value is a non zero value.
1	18.	(Previously Presented)	The system recited in claim 10, wherein the
2	at least one	status bit is set high based	on the value of the least significant bit of the
3	digital value.		

1	19. (Previously Presented) In a pipelined instruction processor that
2	executes instructions including conditional jump instructions, one or more of the
3	conditional jump instructions reading a digital value from memory to determine if the
4	condition of the conditional jump instruction is satisfied, the improvement comprising:
5	a plurality of addressable registers, each of the addressable registers storing a
6	value that includes a digital value and at least one jump status bit;
7	logic to access a current instruction, wherein the current instruction includes an
8	address and a corresponding jump field, the address identifies one of the addressable
9	registers and the corresponding jump field identifies a jump status bit of the at least one
10	jump status bits within the identified addressable register;
11	a jump look-ahead controller for generating a jump look-ahead signal using the
12	address that identifies one of the addressable registers and the jump field that identifies a
13	jump status bit within the identified addressable register, the jump look-ahead signal is a
14	function of the identified jump status bit;
15	tracking logic for tracking the addresses of a predetermined number of previous
16	instructions in the pipelined instruction processor and comparing the addresses of each
17	previous instruction to the address of the current instruction to generate a series of jump
18	disable signals; and
19	conflict detection logic for generating a jump early signal using the jump look-
20	ahead signal and the series of jump disable signals, the jump early signal initiates the
21	conditional jump depending on the values of the jump disable signals.

1	20. (Previously Presented) The pipelined instruction processor as
2	recited in claim 19, wherein each jump status bit is dependent on the digital value stored
3	in the corresponding addressable register.
1	
1	21. (Previously Presented) The pipelined instruction processor as
2	recited in claim 19, further comprising a bit status generator for generating the
3	corresponding jump status bits.
1	22. (Previously Presented) The pipelined instruction processor as
2	recited in claim 19, further comprising a prediction logic block responsive to the jump
3	early signal for implementing a prediction algorithm to predict the conditional jump
4	depending on the values of the jump disable signals.
1	23. (Previously Presented) The pipelined instruction processor as
2	recited in claim 19, wherein the tracking logic includes a queue for sequentially storing
3	pre-determined number of instructions prior to sequentially piping the pre-determined
4	number of instructions through a read stage and decode stage in a pre-fetch pipeline.
1	24. (Previously Presented) The pipelined instruction processor as
2	recited in claim 23, wherein the pre-determined number of instructions are sequentially

6123599349

3	piped through an execution pipeline after being piped through a pre-fetch pipeline, the		
4	execution pipeline includes a write-back stage.		
1			
1	25. (Previously Presented) The pipelined instruction processor as		
2	recited in claim 24, whercin the addressable register is written during the write-back		
3	stage.		
1	26. (Previously Presented) The pipelined instruction processor as		
2	recited in claim 25, wherein the execution pipeline further includes an address generation		
3	stage, a present address stage, an output operand stage, a capture data stage, and an		
4	arithmetic operation stage, all before the write-back stage.		
1	27. (Previously Presented) A method for determine if a condition of		
2	a conditional jump instruction is satisfied in a pipelined instruction processor, the		
3	method comprising:		
4	storing a digital value and one or more jump status bits that are based on the		
5	digital value in one or more of a plurality of address locations in an addressable		
6	memory;		
7	accessing a current instruction, the current instruction having an address and a		
8	jump field, the address identifies a selected address location of the addressable		

9	memory, and the jump field identifies a selected jump status bit of the selected
10	address location;
11	generating a jump look-ahead signal that is a function of the selected jump
12	status bit read from the selected address location of the addressable memory, the
13	identified jump status bit is accessed using the address and the jump field of the
14	current instruction;
15	tracking the addresses of a predetermined number of previous instructions in
16	the pipelined instruction processor and comparing the addresses to the address of the
17	current instruction to generate a series of jump disable signals, and
18	generating a jump early signal using the jump-look ahead signal and the series
19	jump disable signals, the jump early signal initiates a conditional jump depending on
20	the value of the jump disable signals.